HW-2 due 10/06/03

Problems 1, 2 deal with writing, running and testing C++ code. Please follow instructions specified in the document: Submitting C++ Code. Solutions of the remaining problems need to be submitted on paper.

1. Write, run and test the code corresponding to your solutions of problems 2, 4, 5, 6 of HW-1.

2. Overload operator == for the bag class of Lectures 4, 5. Design several tests, which will check that your implementation is correct. Submit your enlarged bag class and your tests in the form explained in Submitting C++ Code.

3. What is the output of the following program?

```cpp
int main()
{
    Bag D;
    D.insert(1);
    D.insert(2);
    D.insert(3);
    D=D+D;
    D=D+D;
    while(D.occurrences(1)!=0)
    {
        D.remove(1);
    }
    cout << D.occurrences(1) << endl;
    cout << D.occurrences(2) << endl;
    cout << D.occurrences(3) << endl;
    cout << D.size()<< endl;
    return 0;
}
```

4. (i) What is the output of the program?

```cpp
#include <iostream>
#include "Point.h"
using namespace std;

int main()
{
    int *ptr_1;
    int *ptr_2;
    ptr_1=new int[10];
    ptr_2=new int[10];
    int i;

    for(i=0;i<10;i++)
    {
        ptr_1[i]=1;
        ptr_2[i]=2;
    }
```


```c
int k;
k=ptr_1[5];
ptr_1[5]=ptr_2[5];
ptr_2[5]=k;

for(i=0;i<10;i++)
    cout << ptr_1[i] << " " << ptr_2[i] << endl;

return 0;
```

(ii) What output do we get if we substitute the block

```c
int k;
k=ptr_1[5];
ptr_1[5]=ptr_2[5];
ptr_2[5]=k;
```

by

```c
int *k;
k=ptr_1;
ptr_1=ptr_2;
ptr_2=k;
```

5. (i) What output do we get in the following program?

```c
void swap(int*& p1,int*& p2)
{
    int *p;
p=p1;
p1=p2;
p2=p;
}

int main()
{
    int i=1;
    int j=2;
    int *ptr1=&i;
    int *ptr2=&j;
    swap(ptr1,ptr2);
    cout << *ptr1 << " " << *ptr2 << endl;
    cout << i << " " << j << endl;

    return 0;
}
```

(ii) What changes if we substitute the line `void swap(int*& p1,int*& p2)` in the above code by `void swap(int* p1,int* p2)`?
6. What happens in the program below if we substitute the line `void allocate(Point* & ptr, int & m)` by `void allocate(Point* & ptr, int m)` and then by `void allocate(Point* ptr, int m)`?

```c++
void allocate(Point* & ptr, int m)
{
    ptr = new Point[m];
    int k;
    for (k = 0; k < m; k++)
    {
        ptr[k].set(cos(2 * 3.14 * k / m), sin(2 * 3.14 * k / m));
    }
}

int main()
{
    int n;
    cout << "Enter a number ";
    cin >> n;

    Point *p;
    allocate(p, n);

    int k;
    for (k = 0; k < n; k++)
    {
        cout << p[k] << endl;
    }

    delete[] p;
    return 0;
}
```